Problems: Unit 5

Problem 1
There are N different elements stored in a direct access structure (for example, a vector with the numbers 1, 2, 3, 4 and 5, or the string abcdefg) and we want to obtain all the different ways of placing those elements, or what is the same, you have to get all the permutations of the N elements. Design an algorithm that uses Backtracking to solve the problem.

Problem 2
Solve the previous problem considering the possibility that the elements repeat themselves (for example, vector 1,2,3,1 or the chain acabada).

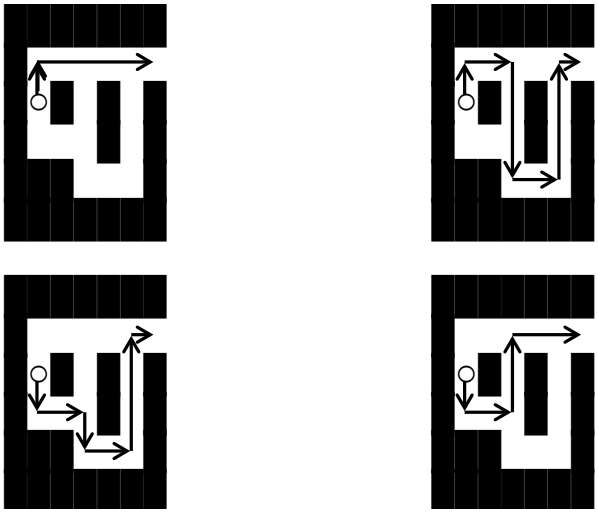
Problem 3
You have a set of N numbers stored in a text string; for example, the string data=1151451. Design an algorithm that by means of Backtracking techniques find, in the most efficient way possible, all the different numbers of N numbers that can be formed with the numbers of the chain without altering its relative order within it.
For example, if N= 4, they are valid numbers 1151, 1511 and 1541, but not 4551 or 5411 that although they can be formed with the digits of the data string imply a reordering.

Problem 4
A board M of size RxC is available (R is the number of rows and C the number of columns) and a chess horse is put in a starting square (posx, posy). The objective is to find, if possible, the way in which the horse must move around the board so that each box is used only once in the course (the 8x8 board always has a solution regardless of where the horse starts). The horse can finish in any position on the board.
A horse has eight possible movements (assuming, of course, that it does not leave the board). A movement between the squares M_{ij} and M_{pq} is valid only if:

- (|p-i|=1) && (|q-j|=2), or if
- (|p-i|=2) && (|q-j|=1),

that is, a coordinate changes two units and the other a single unit.

Problem 5
There is a labyrinth table [1..n, 1..m] with logical values representing a labyrinth.
The TRUE value indicates the existence of a wall (cannot be traversed), while FALSE represents a recordable box.
To move through the labyrinth, you can move horizontally or vertically from one square, but only to an empty square (FALSE). The edges of the table are completely TRUE except one box, which is the output of the labyrinth. Design a Backtracking algorithm that finds all the possible paths that lead to the exit from a certain initial box, if it is possible to leave the labyrinth.



Design a Backtracking algorithm that finds the best possible path that leads to the exit from a certain initial box, if it is possible to leave the labyrinth.

Problem 6
You have the substitution table that appears below

	a	b	c	d
a	b	b	a	d
b	c	a	d	a
c	b	a	c	c
d	d	c	d	b

which is used in the following way: in any string, two consecutive characters can be replaced by the value that appears in the table, using the first character as row and the second character as column. For example, you can change the sequence ca to a b, since M[c, a]=b.
Implement a Backtracking algorithm that, starting from a string of text and using the information stored in a substitution table M, is able to find a way to make the substitutions that allow reducing the text string to a final character, if possible.
Example: With the string text = acabada and the final character = d, a possible form of substitution is the following (the sequences that are replaced are marked for clarity): aca*ba*da → a*ca*cda → abc*da* → *ab*cd → b*cd* → *bc* → d.

Deliverable: a problem to choose among problems 3 and 4 and problem 6.